

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

Docket No. AUS920030548US1

CLAIMS:

What is claimed is:

1. A method in a data processing system for autonomically determining execution flow of a computer program, comprising:

providing a set of hardware registers for identifying a work area for a thread of the computer program, wherein the work area stores thread tracking information for the thread;

copying thread tracking information from the work area to a buffer using the set of hardware registers;

retrieving symbolic data for the thread; and

generating a call sequence of the computer program based on the symbolic data for the thread.

2. The method of claim 1, wherein the set of hardware registers includes a work area register, a work area length register, and a current pointer register.

3. The method of claim 2, wherein the work area register includes a pointer pointing to a beginning of the work area for the thread.

4. The method of claim 2, wherein the work area length register includes one of a size of the work area for the thread or a pointer pointing to an end of the work area for the thread.

Docket No. AUS920030548US1

5. The method of claim 2, wherein the current pointer register includes a pointer pointing to the location of the work area where last thread tracking information is written.

6. The method of claim 1, wherein the thread tracking information for the thread includes a plurality of call stack entries for the thread.

7. The method of claim 6, wherein each of the plurality of call stack entries is written upon detection of one of a method call and a method return for the thread.

8. The method of claim 7, wherein each of the plurality of call stack entries includes an address to and an address from which one of the method call and a method return is executed.

9. The method of claim 8, wherein each of the plurality of call stack entries further includes additional information, and wherein the additional information includes time stamps and performance monitoring counter values.

10. The method of claim 9, wherein the additional information is compressed with the address to and the address from which one of the method call and a method return is executed when each of the plurality of call stack entries is written.

Docket No. AUS920030548US1

12. The method of claim 9, wherein the additional information is compressed with the address to and the address from which one of the method call and a method return is executed when the thread tracking information is copied from the work area to a buffer.

13. The method of claim 1, wherein copying thread tracking information from the work area to a buffer includes copying thread tracking information when a work area overflow occurs.

14. The method of claim 1, wherein the buffer is one of a trace buffer and a consolidated buffer accessible by an application.

15. The method of claim 1, wherein retrieving symbolic data for the thread includes retrieving symbolic data from an indexed symbolic database by searching the indexed symbolic database for symbolic data based on a process identifier for the thread.

16. The method of claim 15, wherein the symbolic data matches the process identifier for the thread and the address of one of a method call and a method return for the thread.

17. The method of claim 1, wherein retrieving symbolic data for the thread includes retrieving symbolic data from one of a directory of the loaded module, a shadow

Docket No. AUS920030548US1

directory, or a loaded module if an address of the loaded module on a disk is known.

18. The method of claim 1, wherein generating a call sequence of the computer program includes associating the retrieved symbolic data with the thread tracking information in the buffer.

19. A data processing system for autonomically determining execution flow of a computer program, the data processing system comprising:

- providing means for providing a set of hardware registers for identifying a work area for a thread of the computer program, wherein the work area stores thread tracking information for the thread;

- copying means for copying thread tracking information from the work area to a buffer using the set of hardware registers;

- retrieving means for retrieving symbolic data for the thread; and

- generating means for generating a call sequence of the computer program based on the symbolic data for the thread.

20. The data processing system of claim 19, wherein the retrieving means comprises:

- searching means for searching an indexed symbolic database for symbolic data based on a process identifier for the thread.

Docket No. AUS920030548US1

21. The data processing system of claim 19, wherein the generating means comprises:

associating means for associating the retrieved symbolic data with the thread tracking information in the buffer.

22. A computer program product in a computer readable medium for determining execution flow of a computer program, the computer program product comprising:

first instructions for providing a set of hardware registers for identifying a work area for a thread of the computer program, wherein the work area stores thread tracking information for the thread;

second instructions for copying thread tracking information from the work area to a buffer using the set of hardware registers;

third instructions for retrieving symbolic data for the thread; and

fourth instructions for generating a call sequence of the computer program based on the symbolic data for the thread.

23. The computer program product of claim 22, wherein the third instruction comprises:

sub-instructions for searching an indexed symbolic database for symbolic data based on a process identifier for the thread.

Docket No. AUS920030548US1

24. The computer program product of claim 22, wherein the fourth instruction comprises:

sub-instructions for associating the retrieved symbolic data with the thread tracking information in the buffer.